Lehrstuhl für Informatik I
— Informationsstrukturen und
wissensbasierte Systeme —
Universität Würzburg

Prof. Dr. H. Noltemeier

Würzburg, 18. Juni 1998

Übungen zur Vorlesung
Praktische Informatik II (SS98)

Musterlösung zum 3. Übungsblatt

Lösung zu Aufgabe 10: (Segmentbaum; lokale Schranken)

Sei $ST(l,r)$ ein Segmentbaum, $l \leq b \leq e-3 < e \leq r$ und $\overline{S}(l,r,b,e) := \min\left\{S, 1+2\left\lceil\log_2 \frac{e-b}{2}\right\rceil\right\}$ mit $S = \lceil\log_2(r-l)\rceil + \lfloor\log_2(r-l)\rfloor - 2$.

Zu zeigen:

Für die Anzahl k der Intervalle, in die das Intervall $[b,e]$ beim Einfügen in $ST(l,r)$ zerlegt wird, gilt: $k \leq \overline{S}(l,r,b,e)$.

Beweis:

Wegen der rekursiven Definition der Operation INSERT kann o.B.d.A. angenommen werden, daß das Intervall $[b,e]$ in $ST(l,r)$ eingefügt wird mit $l \leq b \leq \lfloor\frac{l+r}{2}\rfloor \leq e \leq r$, d.h. das Intervall $[b,e]$ wird möglicherweise auf beide Teilbäume aufgeteilt. Dies entspricht dem Einfügen von $[b,\lfloor\frac{l+r}{2}\rfloor]$ in den linken und von $[\lfloor\frac{l+r}{2}\rfloor,e]$ in den rechten Teilbaum.

Im Folgenden wird dann pro Ebene und Teilbaum höchstens ein Standardintervall alloziert, denn angenommen, es würden o.B.d.A. im linken Teilbaum auf einer Ebene zwei Intervalle $[s_1,t_1]$ und $[s_2,t_2]$ mit $s_1 < t_1 \leq s_2 < t_2$ für $[b,\lfloor\frac{l+r}{2}\rfloor]$ alloziert. Dann kann keiner der beiden zugehörigen Knoten die Teilbaumwurzel sein. Sei $[s,t]$ das Standardintervall des Vaters des zu $[s_2,t_2]$ gehörenden Knotens. Dann ist $[s,t] \subseteq [s_1,\lfloor\frac{l+r}{2}\rfloor] \subseteq [b,\lfloor\frac{l+r}{2}\rfloor]$, und bereits das Intervall des Vaters wäre alloziert worden, im Widerspruch zur Allokation des Intervalls des Sohnes (siehe Algorithmus INSERT).

Die effektive Höhe der betrachteten Teilbäume ist $h_l = \lceil\log_2(\lfloor\frac{l+r}{2}\rfloor - b)\rceil$ bzw. $h_r = \lceil\log_2(e - \lfloor\frac{l+r}{2}\rfloor)\rceil$, da es nur auf den h_l+1 bzw. h_r+1 untersten Ebenen der Teilbäume Standardintervalle gibt, deren Breite kleiner oder gleich $\lfloor\frac{l+r}{2}\rfloor - b$ bzw. $e - \lfloor\frac{l+r}{2}\rfloor$ ist. Daher können höchstens auf diesen Ebenen Allokationen stattfinden.

Wird das Wurzelintervall des linken bzw. rechten Teilbaumes alloziert, so wird dadurch zumindest die Hälfte von $[b,e]$ abgedeckt; genauer gilt $e - \lfloor\frac{l+r}{2}\rfloor \leq \lceil\frac{e-b}{2}\rceil$ bzw. $\lfloor\frac{l+r}{2}\rfloor - b \leq \lceil\frac{e-b}{2}\rceil$, d.h. die effektive Höhe des anderen Teilbaumes ist dann $\lceil\log_2\lceil\frac{e-b}{2}\rceil\rceil$. Folglich werden in ihm höchstens $1 + \lceil\log_2\lceil\frac{e-b}{2}\rceil\rceil \leq 2 + \lceil\log_2 \frac{e-b}{2}\rceil$ Standardintervalle alloziert. Wegen $e-b \geq 3$ sind dies höchstens $1+2\lceil\log_2 \frac{e-b}{2}\rceil$ viele.

Demnach dürfen wir von nun an davon ausgehen, daß die Wurzelintervalle der Teilbäume nicht alloziert werden. Also gilt für die Anzahl der belegten Intervalle im linken bzw. rechten Teilbaum $k_l \leq h_l$ und $k_r \leq h_r$.

Aus Symmetriegründen kann man sich im Folgenden auf den Fall $\lfloor\frac{l+r}{2}\rfloor - b \leq \frac{e-b}{2}$ beschränken. Dann ist

$$h_l \leq \left\lceil \log_2 \frac{e-b}{2} \right\rceil$$

und wegen $\frac{e-b}{2} \leq e - \lfloor\frac{l+r}{2}\rfloor \leq e - b$ gilt

$$h_r \leq \lceil \log_2(e-b) \rceil = \left\lceil \log_2 \left(2 \cdot \frac{e-b}{2}\right) \right\rceil = 1 + \left\lceil \log_2 \frac{e-b}{2} \right\rceil$$

Insgesamt ist also

$$k = k_l + k_r \leq h_l + h_r \leq 1 + 2 \left\lceil \log_2 \frac{e-b}{2} \right\rceil$$

Mit dem Ergebnis $k \leq S$ aus Aufgabe 9 (c) folgt schließlich die zu zeigende Aussage. □

Lösung zu Aufgabe 11: (Prioritätssuchbaum)

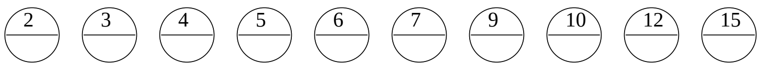
Es sei die Punktmenge

$$S := \{(2,9),(3,4),(4,7),(5,1),(6,3),(7,5),(9,10),(10,2),(12,13),(15,4)\}$$

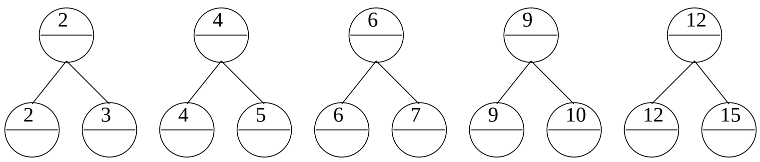
über dem Universum $U = \{2,3,4,5,6,7,9,10,12,15\}$ gegeben.

(a) Erzeugen des Baumes

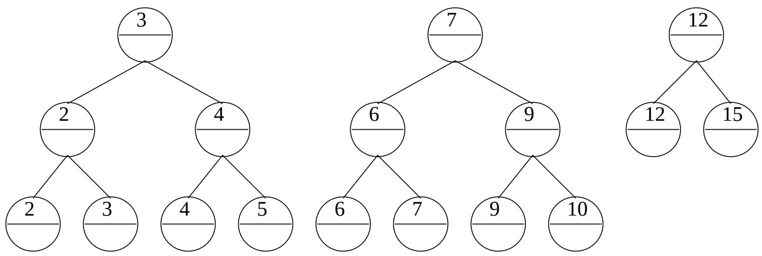
- Preprocessing: Erzeugen des Skeletts
  - Für jeden x-Wert wird ein Blatt erzeugt (Sortierung!)



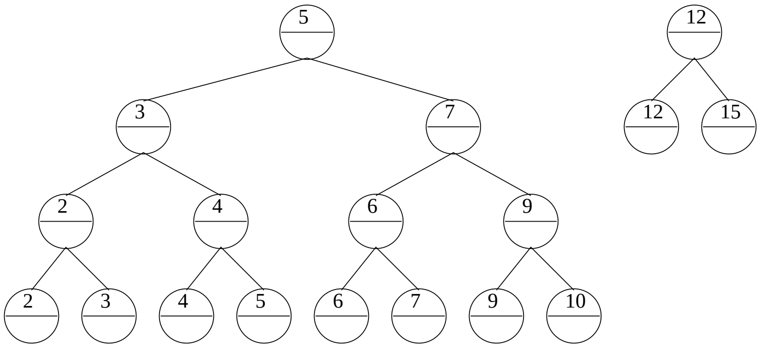
  - Dann faßt man von links nach rechts jeweils zwei Blätter zusammen. Der Wert des linken Blattes wird stets als Splitwert in den Vater geschrieben.



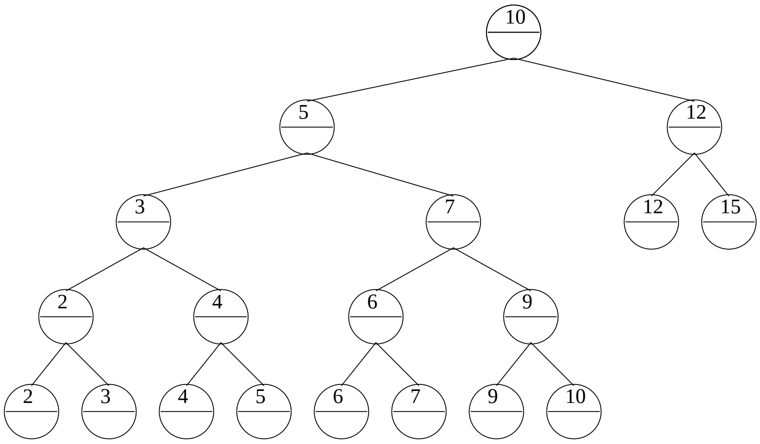
  - Im nächsten Schritt werden die Teilbäume wiederum von links nach rechts zusammengefaßt. Als Splitwert in der neuen Wurzel wird der größte Wert im linken Teilbaum verwendet. Ein Teilbaum bleibt allerdings übrig. Dieser kann frühestens im nächsten Schritt berücksichtigt werden.



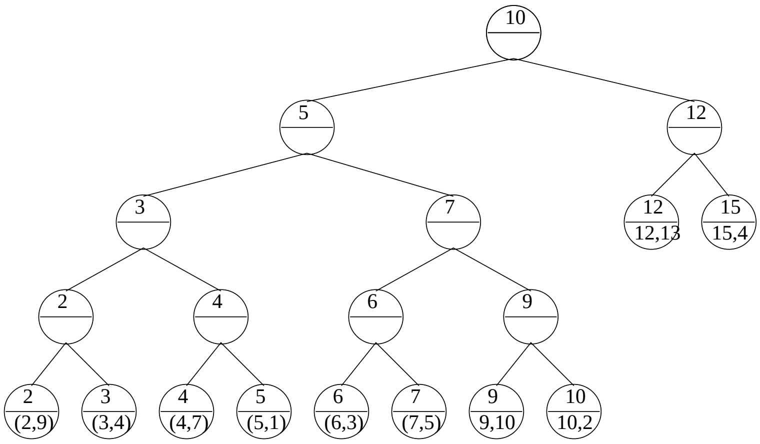
– Auch hier bleibt ein Teilbaum übrig:



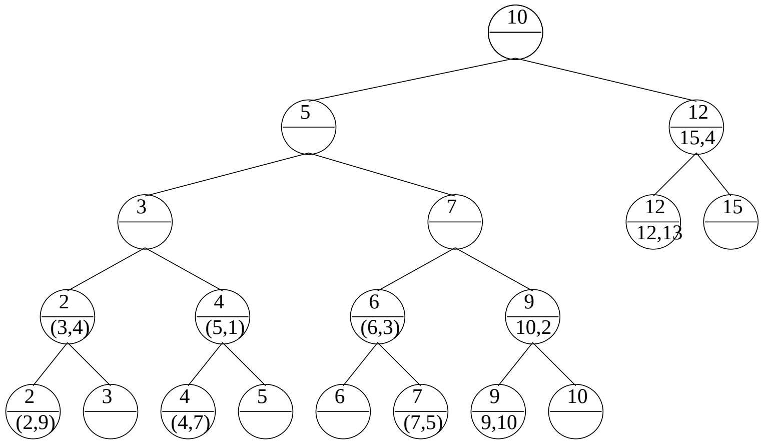
– Das fertige Skelett:



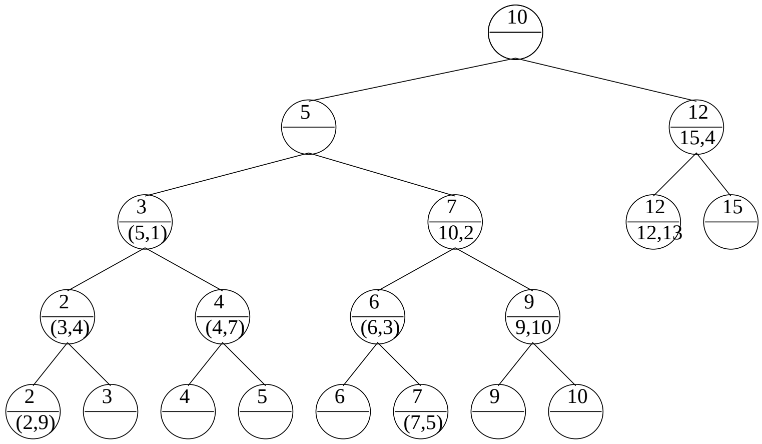
- Einfügen der Werte durch iteratives Bottom-Up-Insert
  – Alle Punkte aus S werden entsprechend ihrem x-Wert in das zugehörige Blatt eingefügt.



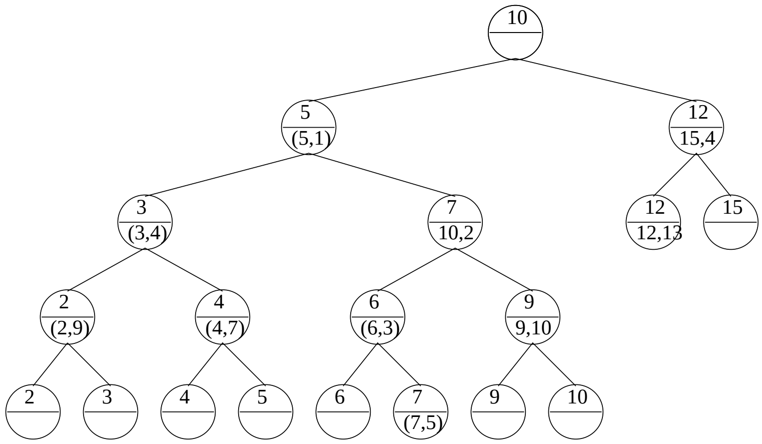
– Von zwei Punkten, die in Blättern mit einem gemeinsamen Vater gespeichert sind, wird der mit dem kleineren y-Wert in den Vater hochgezogen (*bubble-up*).

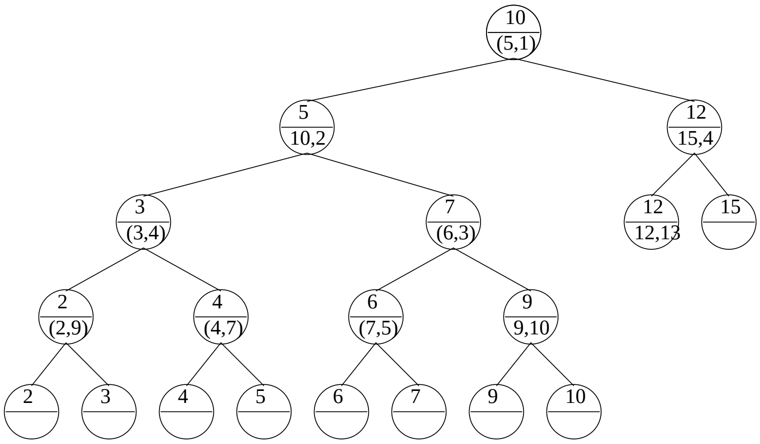


– Wie im vorherigen Schritt werden freie Plätze in der nächsten Ebene belegt, hierbei müssen auf den darunterliegenden Ebenen entstehende Löcher rekursiv aufgefüllt werden.

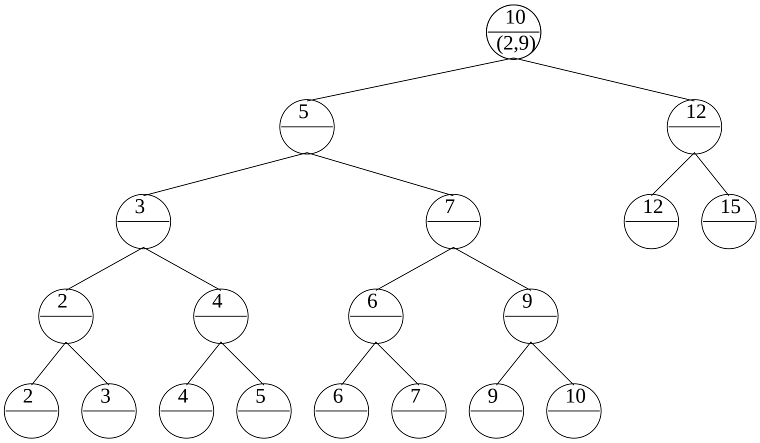


– Fährt man so fort, erhält man wie in den Abbildungen gezeigt schließlich den fertigen Baum.

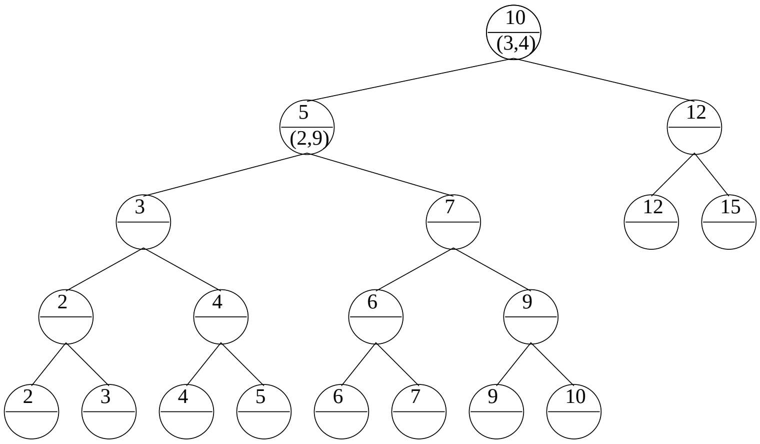




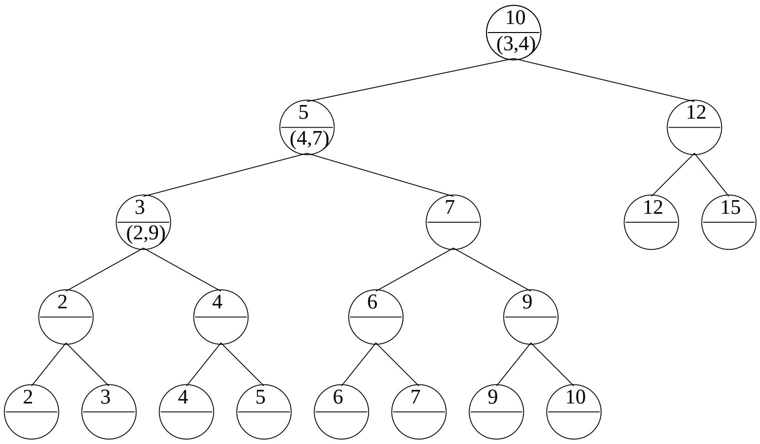
- Einfügen der Menge S in den Baum durch iteratives Anwenden der Operation INSERT
  - Einfügen von $(2,9)$:



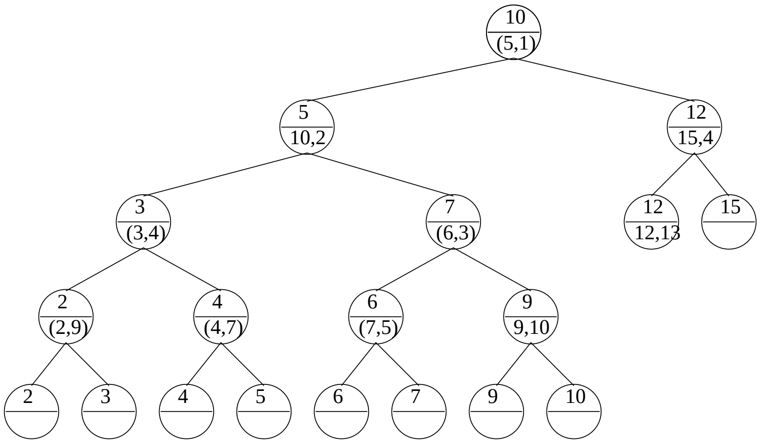
  - Einfügen von $(3,4)$:
    Der in der Wurzel befindliche Punkt $(2,9)$ hat die höhere Priorität. Deswegen wird $(3,4)$ in der Wurzel gespeichert, und dann $(2,9)$ in den Baum eingefügt.



– Einfügen von $(4, 7)$:
Der in der Wurzel befindliche Punkt $(3, 4)$ hat geringere Priorität als $(4, 7)$. Da der Splittwert der Wurzel 10 größer als 4 ist, wird in den linken Teilbaum eingefügt.
Dort hat der Wert $(2, 9)$ höhere Priorität, weshalb $(3, 4)$ in der Wurzel gespeichert wird, und dann $(2, 9)$ in den Baum eingefügt wird.



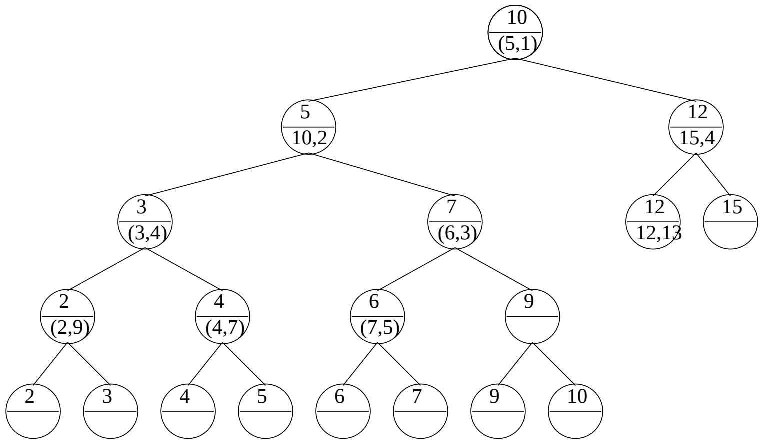
– Fährt man so fort, erhält man den fertigen Baum:



(b) Nun sind die Punkte $(9, 10)$ und $(5, 1)$ zu löschen.

- Löschen von $(9, 10)$:

  Das Löschen von $(9, 10)$ bereitet keine Schwierigkeiten, da in den Knoten im entsprechenden Teilbaum keine weiteren Punkte gespeichert sind.



- Löschen von $(5, 1)$:

  Der Punkt $(5, 1)$ befindet sich in der Wurzel. Man entfernt den Punkt und läßt den Punkt $(10, 2)$ des linken Sohns aufsteigen, da er von beiden Söhnen die geringere Priorität hat. Nun fährt man rekursiv mit dem linken Teilbaum so fort und erhält schließlich folgenden Baum: